



● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

2009 Laurea Leppävaara

Laurea-ammattikorkeakoulu
Laurea Leppävaara

Matriisikoodit ja niiden käyttö opiskelijaruokalassa

Jari Tukiainen
Tuukka Tiilimäki
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Marraskuu, 2009

Jari Tukiainen
Tuukka Tiilimäki

Matriisikoodit ja niiden käyttö opiskelijaruokalassa

Vuosi 2009

Sivumäärä 32

Matriisikoodit eli kaksiulotteiset viivakoodit ovat laajennettuja viivakodeja. Ne pystyvät sisältämään huomattavasti enemmän tietoa kuin tavalliset yksiulotteiset viivakoodit. Tämän ansiosta niillä on enemmän käyttötarkoituksia ja -mahdollisuuksia kuin tavallisilla viivakodeilla.

Opinnäytetyö tehtiin Laurea-ammattikorkeakoulun tietojenkäsittelyn koulutusohjelmassa. Tavoitteena oli tutkia matriisikodeja, niiden käyttömahdollisuuksia sekä tuottaa toimiva matriisikodeja hyödyntävä sovellus.

Työ sisältää kirjallisuusselvityksen lisäksi toiminnallisen osuuden. Kehitimme ohjelman joka esittelee opiskelijaruokala BarLaurean ruoka-annosten ravintoarvoja. BarLaurean asiakas voi hakea kännykällään päivän annoksen ravintotiedot kuvaamalla ruokalaan sijoitetun matriisikoodin.

Työn tuloksena tuotettiin toimiva ohjelma ravintoarvojen tarkastelua varten. Ohjelma hyödyntää Laureassa vuonna 2006 tehtyä "Ateriatietopalvelun kehittäminen ja käyttöönotto BarLaureassa" -opinnäytetyötä. Työhön sisältyi rajapinta ravintoarvojen kyselyä varten laajasta tietokannasta.

Jari Tukiainen
Tuukka Tiilimäki

Matrix codes and their use in a student restaurant

Year	2009	Pages	32
------	------	-------	----

Matrix codes, or two-dimensional bar codes, are extended bar codes. They can contain significantly more information than regular one-dimensional bar codes. Because of this they have more uses and possibilities than regular bar codes.

This thesis was completed in the Information Technology Programme of Laurea University of Applied Sciences. The objective was to research matrix codes, their possible uses and produce a working application that uses matrix codes.

The thesis includes a report and also a functional section. A program was developed that shows the nutritional values of the student restaurant BarLaurea's menu. A customer at BarLaurea can access the nutritional values of the day's meal by taking a picture of a matrix code placed in BarLaurea.

The outcome of the thesis was a functional program for viewing nutritional values. The program makes use of a thesis made in Laurea in 2006 called "Ateriatietopalvelun kehittäminen ja käyttöönotto BarLaureassa" (The development and implementation of a meal information system in BarLaurea). The thesis included an interface for querying nutritional values from a large database.

Key words: Matrix codes, 2D codes, barcodes, Meal Information Service (ATP)

Sisällys

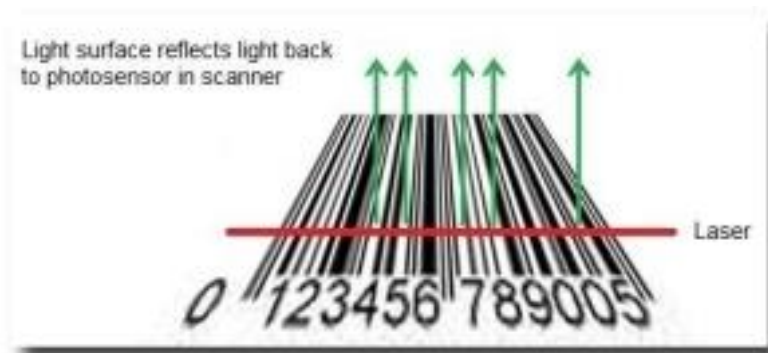
1	Johdanto	5
2	Opinnäytetyön lähtökohdat ja tutkimusmenetelmä	8
2.1	Lähtökohdat	8
2.2	Tutkimusmenetelmä	8
3	Matriisikoodien käyttötarkoituksia	9
3.1	Objektien hyperlinkittäminen	9
3.2	Muita käyttötarkoituksia	10
3.3	Matriisikoodit teollisuudessa	11
4	Matriisikodeja	11
4.1	Semacode	11
4.2	Datamatrix	12
4.3	Shotcode	12
4.4	PDF417	13
4.5	Aztec-koodi	13
4.6	QR-koodi	14
4.7	Microsoft Tag	15
4.8	UpCode	16
5	Matriisikoodien käytännön sovelluksia	17
5.1	HKL:n kaikille pysäkeille Upcode-koodit	17
5.2	Automaattinen paikoitusmaksusovellus	17
5.3	Facebook	17
5.4	Denim Code	18
5.5	Abukuma Chicken Farm	18
6	Matriisikoodien hyödyntäminen opiskelijaruokalassa	19
6.1	Mobiili-ATP:n toteutus	20
6.2	Mobiili-ATP:n käyttö	21
7	Matriisikoodien tulevaisuudennäkymiä (ATOM TAG)	23
8	Tulokset ja johtopäätökset	23
	Sanasto	25
	Lähteet	27
	Kuvat	28
	Liitteet	29

1 Johdanto

Viivakoodit ovat mekanismi tiedon lukemiseen ilman numeroiden tai kirjainten kirjoitusta. Melkein kaikessa kaupasta löytyvässä tavarassa on nykyisin yksilöivä viivakoodi. Viivakoodin alta löytyvä numerosarja on sisällytetty viivakoodiin. Sitä ei lueta koodinlukijalla, vaan se on olemassa, jotta asiakas voi esimerkiksi tuotteen ostettuaan ja jonkun vian löydettyään ottaa yhteyttä asiakaspalveluun ja numerosarjan perusteella yksilöidä tuotteen. Jokapäiväisessä elämässä nähtävät viivakoodit ovat EAN-formaatissa (European Article Number) ja ne on luokiteltu yksiulotteisiksi eli 1D-viivakoodiksi. (QR-codes explained 2009.)

Vuonna 1952 Joseph Woodland, Jordin Johanson sekä Bernard Silver saivat patentoitua systeemin rautatievaunujen tunnistamiseen. Junavaunun kylkeen laitettiin leima, jonka muste hohti lukijalaitteen uv-valossa. Viivakoodit tulivat yleiseen käyttöön vuonna 1966 UPC-koodien (Universal Product Code) muodossa, joka oli lyhyempi kuin nykyisin käytössä oleva EAN-13. (QR-codes explained 2009.)

Monet luulevat, että viivakoodin lukulaite lukee mustien palkkien perusteella. Todellisuudessa lukeminen tapahtuu valkoisten alueiden perusteella, jotka heijastavat valoa takaisinpäin lukulaitteeseen, jonka sensorit tulkitsevat koodin valon perusteella. Viivakoodin korkeudeksi riittäisi vain 1 mm, mutta koodin lukeminen on huomattavasti helpompaa, kun se on korkeampi. (QR-codes explained 2009.) Kuvassa 1 havainnollistetaan viivakoodin lukemista.



Kuva 1: Viivakoodi ja sen lukeminen

Matriisikoodi on kaksiulotteinen viivakoodi, joka sisältää mustia ja valkoisia "soluja" eli moduuleita yleensä järjestettynä joko neliö- tai suorakaiteen muotoiseksi. Koodattava tieto voi olla tekstiä tai muuta tietoa. Yleinen tiedon koko on muutamista tavuista 2 kilotavuun. Koodatun tiedon pituus riippuu käytetystä symboliulottuvuudesta. Virheenkorjauskoodeja käytetään lisäämään matriisikoodien "vahvuutta": vaikka ne olisivatkin osittain vaurioituneita, ne voidaan silti lukea. (Datamatriisit 2009.)

Matriisikoodit ovat yleensä neliön muotoisia ja ne koostuvat soluista eli moduuleista, siis pienistä elementeistä, jotka edustavat bittejä. Tilanteen mukaan valkoinen solu vastaa nollaa ja musta ykköstä tai päinvastoin. Jokainen matriisi koostuu kahdesta vierekkäisestä reunasta L-kirjaimen muodossa, (tätä kutsutaan "etsintäkuvioksi") ja kahdesta muusta reunasta, jotka koostuvat sekä mustista että valkoisista soluista, (tätä kutsutaan "ajoituskuvioksi"). Näiden reunojen sisällä on riveittäin ja sarakkeittain soluja, jotka sisältävät koodattua tietoa. Etsintäkuviota käytetään paikallistamaan ja suuntaamaan symboli samalla, kun ajoituskuvio kertoo rivien ja sarakkeiden määrää symbolissa. Kun enemmän tietoa on koodattu symboliin, solujen määrä (rivit ja palkit) nousee. (Datamatriisit 2009.) Kuvassa 2 on esimerkki UpCode-matriisikoodista.



Kuva 2: UpCode-matriisikoodi

Matriisikoodoja kuten muitakin Open Source -koodoja, esimerkiksi perinteistä yksiulotteista viivakoodia, voidaan lukea nykyään mobiililaitteilla lataamalla asiaankuuluva ohjelmisto laitteeseen. Siinä missä perinteinen viivakoodi sisältää vain numerosarjan, joka vastaa jotakin kaupan tietojärjestelmän hintaa, matriisikoodiin voidaan sisällyttää esimerkiksi nettilinkkejä, jotka ohjaavat puhelimen automaattisesti jollekin www-sivulle. Esimerkiksi Japanissa matriisikoodit ovat yleisempiä ja niitä saatetaan painaa hyvinkin isoina, jotta koodin voi lukea kaukaa, esimerkkinä kuva 3. (QR-koodi 2009.)



Kuva 3: Kaukaa kuvattava matriisikoodi

Matriisikoodit koostuvat soluista eli moduuleista, joihin voidaan koodata 3116 eri merkkiä ASCII-merkistöstä. Matriisikoodi on siis kokoelma tietoa alueita, jotka sisältävät järjesteltyjä moduuleita. Suuret symbolit sisältävät useita alueita. Jokainen alue rajoittuu etsintäkuvioon, ja sitä ympäröi marginaali (quiet zone border). Moduulit voivat olla pyöreitä tai neliöitä - standardissa ei määritellä tarkkaa muotoa. (Datamatriisit 2009.)

Matriisikoodilla on parillinen määrä rivejä ja parillinen määrä sarakkeita. Suurin osa symboleista on neliöitä vaihdellen 10x10-koosta 144x144-kokoon. Jotkut symbolit ovat suorakaiteen muotoisia vaihdellen koosta 8x18 kokoon 16x48. Kaikki symbolit, jotka käyttävät ECC200- virheenkorjausta, voidaan tunnistaa oikean yläkulman moduulista, joka on aina sama kuin taustaväri (0). ECC200 tarjoaa kehittyneitä virheentarkistusta ja korjausalgoritmeja, kuten Reed-Solomon. ECC200 mahdollistaa täydellisen tiedon palautuksen, vaikka symbolista olisi vaurioitunut 30 %. Tiedon palautus olisi mahdotonta tavallisilla viivakodeilla. (Datamatriisit 2009.)

2 Opinnäytetyön lähtökohdat ja tutkimusmenetelmä

2.1 Lähtökohdat

Opinnäytetyö liittyy Laureassa tehtyyn aikaisempaan opinnäytetyöhön. Mikko Lassilan vuonna 2006 tekemä ”Ateriatietopalvelun kehittäminen ja käyttöönotto” toimii pohjana työllemme, koska se tarjoaa rajapinnan ravintoarvojen kyselyä varten Aromin tietokannasta.

Ateriatietopalvelu on sovellus, jonka avulla ruokalan tai ravintolan asiakkaat voivat tutkia ruokalassa tarjottavien ruokalajien ravintoainesisältöjä sekä raaka-ainetietoja internetin välityksellä. Asiakkaat voivat myös koota tarjottavista ruokalajeista haluamansa kaltaisen ateriakokonaisuuden eli virtuaalitarjottimen ja tarkastella syömiensä ruokien ravintoarvokertymiä ja energiajakaumia kokonaisuutena. BarLaurea on Laurea-ammattikorkeakoulun opiskelijaruokala ja ravintola, joka toimii opiskelijavetoisesti. Ateriatietopalvelu saa ravintosisältötiedot BarLaurean tuotantojärjestelmä Aromiin tallennetuista resepteistä. (Lassila 2006, 5 - 6.)

Aromi on BarLaurean WM-DataNovolta hankittu ammattikeittiöiden tuotannonhallintajärjestelmä, jota voidaan käyttää ruokapalvelutoiminnan hallintaan tuotanto-, materiaali- ja myyntitoimintojen osalta. (Lassila 2006, 14.)

Laboratoriokoordinaattori Julius Tuomisto Laurea Leppävaaran tietoliikennelaboratoriosta ehdotti opinnäytetyömme aiheeksi koulussamme käytössä olleen ateriatietopalvelun mobiiliversiota. Ateriatietopalvelu oli tarkoitettu tavallisella www-selaimella käytettäväksi. Tämä työ lisää ateriatietopalvelun käytettävyyttä tarjoamalla mobiilivaihtoehdon.

Ateriatietopalvelu tarjoaa WebService-tyyppisen rajapinnan kyselyiden suorittamiseksi Aromin tietokantaan. Matriisikoodi tulisi ohjaamaan puhelimen haluttuun osoitteeseen. Osoiteriviltä suoritettaisiin kysely Aromin tietokantaan ja näin saataisiin ravintoarvotietoja mahdollisimman vähällä käyttäjäsyötteellä.

Työn tavoitteena ei ole kuitenkaan pelkkä ateriatietopalvelun mobiiliversio, vaan myös selvitys matriisikodeista ja niiden käyttömahdollisuuksista. Matriisikodeista ei löydy aikaisempaa opinnäytetyötä, ja aihetta käsittelevää kirjallisuutta on vähän.

2.2 Tutkimusmenetelmä

Opinnäytetyö on toteutettu konstruktivisen tutkimusmenetelmän mukaan. Menetelmä luo pohjan työn toteuttamiselle. Koska hyödynnämme jo olemassa olevaa tietoa, työ on

innovatiivinen.

Innovaatio on uudistus, jonka toivotaan tuottavan käyttäjälleen hyötyä. Innovaation toteuttaminen, tuottaminen ja laatiminen poikkeaa ns. perustutkimuksesta, jossa pyritään selvittämään ilmiötä sinänsä. Konstruktiviselle tutkimukselle on luonteenomaista uuden todellisuuden rakentaminen olemassa olevan (tutkimus)tiedon pohjalta. (Järvinen 2000, 102.)

Innovaatiot perustuvat tiettyjen resurssien uudelleen käyttöön. Jos oletamme, että materiaalit, ihmiset ja tiedot ovat erilaisia resursseja, voimme puhua teknisistä, sosiaalisista ja tiedollisista innovaatioista. Esimerkiksi uusi laite on tekninen, uusi organisaatiomuoto on sosiaalinen ja uusi algoritmi on tiedollinen innovaatio. (Järvinen 2000, 102.)

Marchin ja Smithin (1995) mukaan suunnittelutieteen tutkimusten tulokset ovat neljäntyyppisiä: käsitteistöjä, malleja, metodeja ja realisointoja. He määrittelevät ne seuraavasti: Käsitteistö muodostaa tutkimusaiheen sanaston. Malli on joukko proposiioita tai lauseita, jotka ilmaisevat käsitteiden väliset suhteet. Metodi on joukko askelia (algoritmi tai ohjeisto), joita käytetään suorittamaan tehtävä. Realisointi on innovaation toteutus ympäristössään. Innovaation toteutukset operationalisoivat Marchin ja Smithin mukaan käsitteistöjä, malleja ja metodeja. (Järvinen 2000, 104.)

Oman tutkimuksemme tuloksena on realisointi eli innovaation toteutus. Käytämme hyväksi jo olemassa olevaa teknologiaa eli ateriatietopalvelun rajapintaa Aromin tietokantaan. Hyödynnämme sitä uudella tavalla luomalla mobiilisovelluksen, joka toimii rajapinnan varassa.

3 Matriisikoodien käyttötarkoituksia

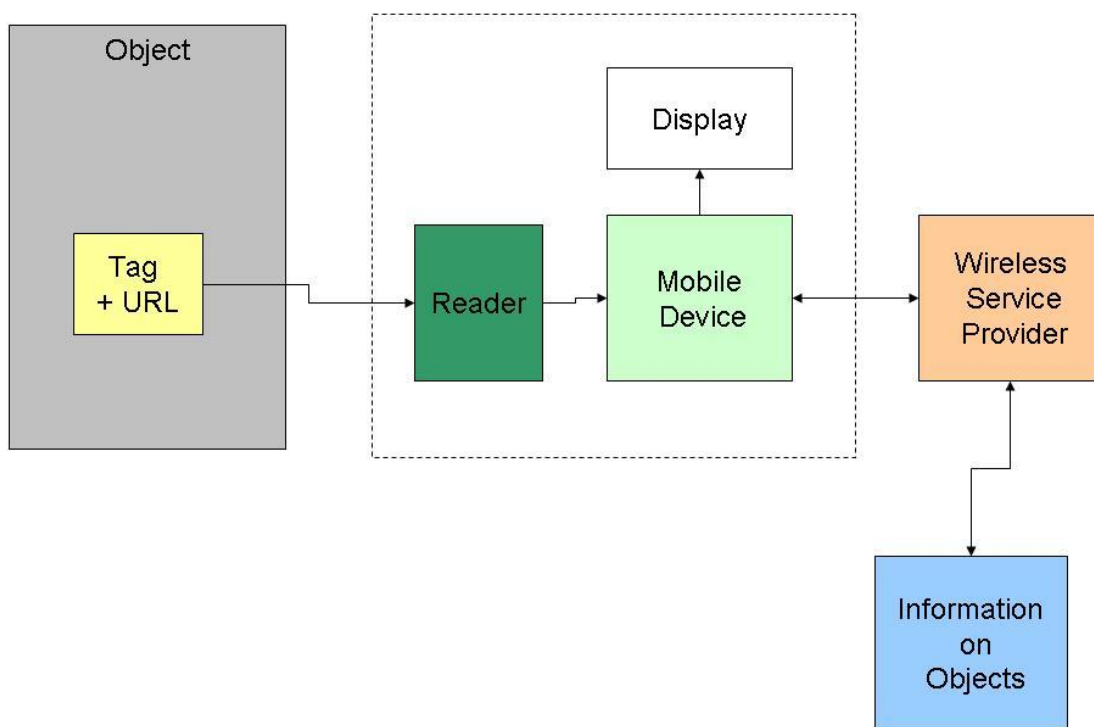
Matriisikoodiin voidaan sisällyttää tietoa, joka esimerkiksi tarjoaa tuotteen tunnistustietoja tai sisältää www-osoitteen. Mobiililaitte voi ohjata käyttäjän internetsivulle, kun käyttäjä kuvaa matriisikoodin mobiililaitteella. Tätä kutsutaan objektien hyperlinkittämiseksi.

3.1 Objektien hyperlinkittäminen

Objektin hyperlinkittäminen tarkoittaa fyysisten esineiden tai kohteiden liittämistä internetiin. Liitos tapahtuu erilaisten URL-osoitteen sisältävien tunnisteiden eli matriisikoodin avulla. Tunniste luetaan mobiililaitteella, joka sitten ohjautuu lisätietoa tai palveluita tarjoavalle sivulle. (Object Hyperlinking 2009.) Seuraavassa listassa esitellään objektin hyperlinkittämisen osat:

1. Fyysinen tai virtuaalinen tunniste objektien ja sijaintien tunnistamiseen
Pieniä fyysisiä tunnisteita voi olla vaikea havainnoida, joten niitä täytyy liittää johonkin huomiota herättävään. Vaihtoehtoisesti koodeja voi painaa suurikokoisina, jolloin ne kiinnittävät huomion sellaisenaan.
2. Tunnisteiden lukemiseen tai löytämiseen tarvittava keino
3. Mobiililaite
4. Tarvittava ohjelmisto mobiililaitteeseen
5. Langaton verkko, kuten 3G, tiedon siirtämiseen palvelimelta mobiililaitteeseen
6. Informaatio jokaisesta linkitetystä objektista, tyypillisesti WWW-sivuna
7. Näyttölaite, eli tyypillisesti kamerakännykän näyttö.
(Object Hyperlinking 2009.)

Kuvassa 4 havainnollistetaan hyperlinkitystä kaaviolla.



Kuva 4: Objektien hyperlinkittäminen

3.2 Muita käyttötarkoituksia

Matriisikoodilla on monia muita käyttötarkoituksia objektien hyperlinkittämisen lisäksi. Suosituin sovellus matriisikoodille on pienten tavaroiden merkitseminen. Koodi on luettavissa 2-3 mm² kokoiselta alueelta ja koodia voi lukea jopa vain 20 % kontrastisuhteella. Matriisikoodi on periaatteessa pienennettävissä tai suurennettavissa loputtomiin. Esimerkiksi 300 mikrometrin kokoisia koodeja on kaiverrettu laserilla 600 mikrometrin kokoiseen

silikonipohjaiseen laitteeseen. Rakennusten seiniin on kiinnitetty useiden metrien läpimittaisia koodeja, jotka on tarkoitettu kaukaa luettavaksi. Ainoana rajoituksena on merkitsemis- ja lukulaitteiden tarkkuus. (Datamatriisit 2009.)

Matriisikoodit ovat yleistymässä painotuotteisiin, esimerkiksi etiketteihin ja kirjeisiin. Koodi voidaan lukea nopeasti skannerilla, mikä mahdollistaa tuotteen seurannan, esimerkiksi milloin paketti on saapunut vastaanottajalle. (Datamatriisit 2009.)

3.3 Matriisikoodit teollisuudessa

Teollisuuden tarpeisiin matriisikoodeja voidaan merkitä suoraan komponentteihin. Koodit voidaan merkitä monilla tavoin, yleisimpinä mustesuihkutus, lasermerkintä, ja elektrolyyttinen kemikaalikaiverrus (electrolytic chemical etching, ECE). Näillä metodeilla saadaan pysyvä jälki, jonka tulisi kestää koko komponentin eliniän. Electronic Industries Alliance (EIA) suosittelee matriisikoodeja pienten elektroniikkakomponenttien merkitsemiseen. (Datamatriisit 2009.)

Matriisin generoinnin jälkeen koodi yleensä varmistetaan erikoiskameralla ja ohjelmalla. Tämä varmistus takaa, että koodi sopii yhteen asiaankuuluvien standardien kanssa, ja varmistaa, että se on luettavissa koko komponentin eliniän. Kun komponentti otetaan käyttöön, matriisi voidaan lukea lukulaitteella joka dekoodaa sen ja sitä voidaan käyttää useaan tarkoitukseen, kuten osan jäljitykseen ja inventaarionhallintaan. (Datamatriisit 2009.)

Matriisikoodit ovat osa monen teollisuuden jäljitettävyyshanketta, erityisesti ilma- ja avaruusteollisuuden parissa, jossa laadunhallinta on tiukkaa ja väärennöksenpaljastus on tärkeää. Matriisikoodit tunnistavat yksityiskohtia ko. komponentista, sisältäen valmistajatunnisteen, osanumeron ja ainutkertaisen sarjanumeron. Yhdysvaltain puolustusministeriö on valinnut matriisikoodit pakolliseksi tunnisteksi tiettyihin hankintoihinsa. Merkintöjä tulee löytyä aseiden osista suurien järjestelmien kriittisiin osiin. (Datamatriisit 2009.)

4 Matriisikoodeja

4.1 Semacode

Semacode on kanadalainen yritys jonka tuotteena on pääasiassa kamerakännykällä luettavat matriisikoodit, jotka avaavat käyttäjälleen verkko-osoitteen. Semacode-koodit ovat ISO/IEC 16022 -standardin mukaisia. Semacode-yrityksen www-sivuilla mainitaan sen olevan avoin systeemi. Semacode-koodien luominen on yksityiselle henkilölle täysin maksutonta ja

rajoittamatonta. Semacode-koodin käyttötarkoitukset ovat hyvin samankaltaisia kuin UpCode-koodin. (Semacode 2009.) Kuvassa 5 esitellään Semacode-koodi.



Kuva 5: Semacode-koodi

4.2 Datamatrix

Datamatrix-koodi on kaksiulotteinen matriisikoodi, joka koostuu mustista ja valkoisista ruuduista, jotka muodostavat neliön muotoisen symbolin. Yleensä se sisältää tietoa muutamasta bitistä kahteen kilobittiin. Tiedon määrän mukaan symbolin ulottuvuus vaihtelee. Datamatrix-koodi voi sisältää jopa 2335 alfanumeerista merkkiä. Datamatrix-koodia käytetään paljon tuotepakkauksissa, ja lukemiseen käytetään skanneria eli koodinlukijaa. (Datamatriisit 2009.) Kuvassa 6 esitellään laajennettu neliosainen Datamatrix-koodi.



Kuva 6: 4-osainen Datamatrix-koodi

4.3 Shotcode

Shotcode on ympyränmuotoinen viivakoodi, joka on kehitetty Cambridgen yliopistossa vuonna 1999. Shotcode-koodi toimii siten, että symbolin keskipisteestä mitattu etäisyys tietorenkaisiin kertoo symboliin varastoidun tiedon, joka on käytännössä koodin määrittämä www-osoite. Shotcode-koodi on tarkoitettu luettavaksi kameralla, mukaan lukien

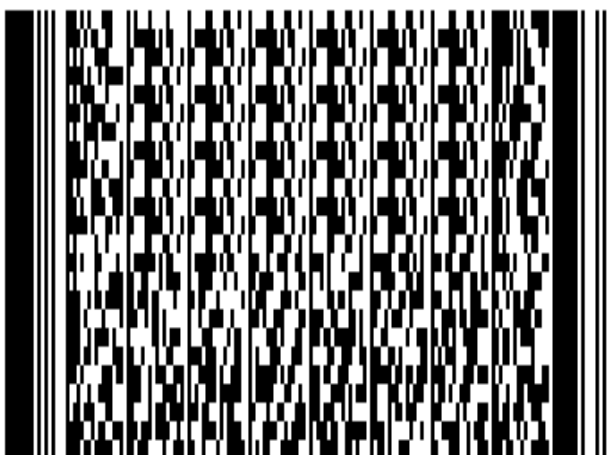
kännykkäkamerat sekä web-kamerat. (Shotcode 2009.) Kuvassa 7 on esimerkki Shotcode-koodista.



Kuva 7: Shotcode-koodi

4.4 PDF417

Kyseessä on logistiikassa ja tunnistekorteissa käytössä oleva pinottu lineaarinen viivakoodisymboli. Koodi on kehitetty vuonna 1991. PDF417-koodi voidaan sisällyttää toinenkin koodi, joka mahdollistaa entistä suuremman määrän tietoa yhdestä symbolista. (PDF417 2009.) PDF417-koodia käytetään myös esimerkiksi KLM-lentoyhtiön lentolipuissa. Esimerkki kuvassa 8.



Kuva 8: PDF417-koodi

4.5 Aztec-koodi

Kyseessä on vuonna 1997 julkaistu tyypillinen matriisikoodi. Koodissa on keskellä keskipiste, ja koodi muodostuu kehistä keskipisteen ympärillä. Koodiin varastoidun tietomäärän

kasvaessa kehien määrä kasvaa ja koodin koko laajenee. Tätä koodia käytetään esimerkiksi juna- ja lentolipuissa, ja koodi voidaan lukea matkustajan matkapuhelimen näytöltä lukulaitteella. (Aztec 2009.) Kuvassa 9 on esimerkki Aztec-koodista.



Kuva 9: Aztec-koodi

4.6 QR-koodi

QR-koodi on kaksiulotteinen viivakoodi, joka on kehitetty japanilaisessa Denso-Wave nimisessä yhtiössä 1994. Esimerkki QR-koodista on kuvassa 10. QR on lyhenne sanoista Quick Response. Kuten nimi kertoo, koodi on luettavissa nopeasti. Tyypillisesti se onnistuu lähes millä tahansa matkapuhelimella, jossa on kamera. QR-koodi on hyvin yleinen Japanissa. Se on käytössä laajalti monissa tuotteissa ja on suosituin matriisikoodityyppi. (QR-koodi 2009.)

QR-koodi, jota on alun perin käytetty osien seurantaan ajoneuvojen valmistusprosessissa, on nykyään käytössä huomattavasti laajemmin. Esimerkkinä ovat kaupalliset seurantasovellukset sekä matkapuhelimien käyttäjille suunnatut hyötyohjelmat. Lehdissä, kylteissä, linja-autoissa ja käyntikorteissa olevat QR-koodit ohjaavat käyttäjän [www-osoitteeseen](#) tämän luettua koodin matkapuhelimensa kameralla. QR-koodeja voi luoda myös itse niitä muodostavalla [www-sivulla](#). (QR-koodi 2009.)



Kuva 10: QR-koodi

4.7 Microsoft Tag

Microsoft Tag -teknologia tunnetaan myös nimellä High Capacity Color Barcode. Matriisit rakentuvat värillisistä kolmioista. Käytettävissä on neljän tai kahdeksan värin matriisi riippuen siitä kuinka paljon tietoa halutaan sisällyttää, mutta myös matriisin sekä kolmioiden kokoa vaihtamalla voidaan vaikuttaa matriisin tiedonvarastointikykyyn. Tämän tyyppiseen matriisiin mahtuu tietoa huomattavasti enemmän kuin perinteisiin matriisikodeihin kuten QR-koodiin. Microsoft Tag käyttää oletuksena 4-väristä systeemiä 5x10 ruudukolla, johon saadaan mahtumaan 13 tavun edestä tietoa. (Microsoft Tag 2009.)

Eri käyttötarkoituksiin voidaan käyttää erikokoisia matriiseja kuten muissakin kaksiulotteisissa koodissa. Esimerkiksi perinteinen web-osoitteen sisältävä Microsoft Tag -matriisikoodi voi kuitenkin olla huomattavasti pienikokoisempi kuin perinteisemmät mustavalkoisia neliöitä sisältävät matriisikoodit. Microsoft Tag -teknologia käyttää Reed-Solomon -virheenkorjausmetodia, jonka ansiosta osittain vahingoittunut tai puutteellinen matriisi voidaan lukea oikein. (Microsoft Tag 2009.) Kuvassa 11 on kokoelma erilaisia Microsoft Tag-koodeja.



Kuva 11: Microsoft Tag -koodeja

Microsoft Tag -koodit voivat myös yhdistää tuotemerkin ja koodinlukemisen yhteen. Tyypillisissä matriisikoodissa täytyy varata tilaa itse koodille, mutta Microsoft Tag -koodin voi yhdistää esimerkiksi yrityksen logoon tai muuhun huomiota herättävään kuvaan. (Microsoft Tag 2009.) Kuvassa 12 verrataan Microsoft Tag-koodin kokoa johtaviin QR- ja Datamatrix-koodeihin.



Kuva 12: Microsoft Tag -koodi verrattuna QR- ja Datamatrix-koodeihin

4.8 UpCode

UpCode on suomalaisen UpCode -yrityksen kehittämä matriisikoodi. (UpCode 2009.) Se toimii kuten moni muukin koodi: se luetaan mobiililaitteella ja vie käyttäjän avulla esimerkiksi nettisivulle, lähettää tekstiviestin tai soittaa tiettyyn numeroon. UpCode-koodia käyttää esimerkiksi Helsingin kaupungin liikennelaitos, jonka pysäkeiltä löytyy UpCode-koodeja, jotka kuvaamalla pääsee aikatauluihin. Myös Kiinteistömaailma käyttää UpCode-koodeja esitteissään ja käyntikorteissaan, kuvaamalla koodin pääsee esimerkiksi myytävän asunnon esittelysivulle. UpCode-koodin käyttö eroaa esimerkiksi QR-koodista siten, että QR-koodiin itseensä on sisällytetty osoite mihin puhelin halutaan ohjata, kun taas UpCode-koodi ohjautuu palvelimelle josta tarkistetaan mihin kyseinen koodi ohjaa. Kuvassa 13 on esimerkki UpCode-koodista.



Kuva 13: UpCode-koodi

5 Matriisikoodien käytännön sovelluksia

5.1 HKL:n kaikille pysäkeille Upcode-koodit

Helsingin kaupungin liikennelaitos lisää kaikille joukkoliikennepysäkeille yksilölliset Upcode-koodit, joilta kameralla ja internetiselaimella varustettu puhelin pystyy hakemaan kyseisen pysäkin aikataulut puhelimeen automaattisesti. (Ojanperä 2008.)

Paperisiin aikatauluihin verrattuna matkustaja saa sähköisen palvelun kautta osalle linjoista reaaliaikaisen pysäkkiaikataulun sekä liikenteen mahdolliset poikkeustiedotteet. (Ojanperä 2008.)

5.2 Automaattinen paikoitusmaksusovellus

UpCode-yritys voitti Piilaaksossa pidetyn mobiilisovellusten kilpailun yrityssarjan automaattisella paikoitusmaksusovelluksella. Sovellus käyttää älytunnistetta, jonka avulla autoilija voi maksaa paikoitusmaksunsa helposti omalla matkapuhelimellaan. (Lahti 2008.)

Kun puhelimen kameralla on otettu kuva UpCode-koodista, matkapuhelin hakee automaattisesti yhteyden autoilijan omaan internet-pankkiin, jossa voi maksaa paikoitusmaksun. Vaihtoehtoisesti maksun voi suorittaa tekstiviestillä. Lisäksi puhelin muistuttaa, kun parkkiaika on loppumassa. (Lahti 2008.)

5.3 Facebook

Facebook kehitti matriisikoodeja hyödyntävän Add to Friends -paidan, kuten kuvassa 14. T-paidassa on painettuna QR-koodi, jonka lukeminen ohjaa lisäämään kyseisen henkilön ystäväksi Facebook-sivustolle. Paidan pystyy tilaamaan internetistä Facebook-sivustolta. (Facebook 2009.)



Kuva 14: Add to Friends -T-paita QR-koodilla

5.4 Denim Code

Ranskalainen Denim Code -merkki painaa farkkuihinsa matriisikoodeja. Valittavana on yli 200 eri koodia miehille ja naisille. Kun puhelimen kameralla on otettu kuva koodista, matkapuhelin hakee automaattisesti yhteyden Denim Code -internetsivulle, jossa voi katsella videoita sekä kuunnella musiikkia. (Denim Code 2009.) Kuvassa 15 on esimerkki Denim Code -farkuista.



Kuva 15: Housut, joissa matriisikoodi

5.5 Abukuma Chicken Farm

Abukuma Chicken Farm -yritys Japanissa on asentanut kameras kanalaan. Kun asiakas ostaa paketin kyseisiä kananmunia, voi hän lukemalla QR-koodin paketin kyljestä siirtyä suoraan lähetykseen kanalasta, jossa kyseiset munat on munittu. (Abukuma 2009.) Kuvassa 16 on kuva kanalasta ja kuvassa 17 esimerkki QR-koodin sisältävästä kananmunapaketista.



Kuva 16: Abukuma Chicken Farm -kanala



Kuva 17: Abukuma Chicken Farm -kananmunia

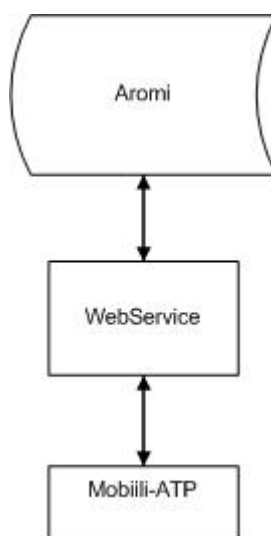
6 Matriisikoodien hyödyntäminen opiskeljaruokalassa

Suunnittelimme ja toteutimme mobiilisovelluksen, joka hyödyntää matriisikoodeja. Sovellus, jonka nimesimme Mobiili-ATP:ksi, eli Mobiili-ateriatietopalveluksi, hyödyntää Mikko Lassilan "Ateriatietopalvelun kehittäminen ja käyttöönotto BarLaureassa" -opinnäytetyötä. Lassila sai työnsä valmiiksi vuonna 2006 ja ateriatietopalvelu oli käytössä tammikuuhun 2008 saakka. BarLaurean henkilökunnalla ei riittänyt aikaa pitää ateriatietopalvelun ruokalistoja ajan tasalla.

Mobiili-ATP:n yhteydessä käytettäväksi matriisikoodiksi valittiin suomalainen UpCode-koodi. Päädyimme valintaan siksi, että UpCode-koodiin ei itseensä sisälly haluttua www-osoitetta, vaan se lähettää kyselyn UpCoden palvelimelle ja tätä kautta selvittää mihin osoitteeseen puhelimen www-selain halutaan ohjata. Tämä mahdollistaa sen, että koodia ei tarvitse

koskaan muuttaa, vaan se päivitetään UpCoden palvelimilta vastaamaan päivän ateriaa. Esimerkiksi QR-koodin kanssa tätä mahdollisuutta ei olisi, ja koodi olisi erilainen joka päivä.

Ateriatietopalvelu sisältää WebService-tyyppisen rajapinnan, jonka avulla Mobiili-ATP pystyy suorittamaan kyselyitä Aromin tietokannasta. WebService on ohjelmistojärjestelmä joka mahdollistaa keskenään yhteensopivien tietokoneiden välisen vuorovaikutuksen tietoverkon yli. Kysely lähetetään Mobiili-ATP:n osoiteriviltä, josta se välittyy WebServiceen, joka suorittaa kyselyn Aromin tietokannasta ja palauttaa sen WebServicen kautta Mobiili-ATP:n käyttäjälle, kuten kuvassa 18.



Kuva 18: Kyselyiden suoritus WebServicen kautta Aromista

6.1 Mobiili-ATP:n toteutus

Mobiili-ATP toteutettiin Microsoft Visual Studiolla C# / .NET -ympäristössä. Se on sivu, jolle lähetetään osoiteriviltä kyselyparametri, ja ohjelma palauttaa halutut tiedot selkeästi kahteen taulukkoon. Jotta ravintoarvot saadaan mahdollisimman vaivattomasti käyttäjän kännykkään, hyödynnämme kyselyssä matriisikoodia, jonka kuvaamalla kysely suoritetaan automaattisesti. Tämä vaatii ylläpitäjältä päivittäin UpCode-linkin päivitystä vastaamaan päivän ateriaa. Käytännössä se tarkoittaa, että ylläpitäjä muuttaa osoitteen, johon UpCode-koodi puhelimen ohjaa. Osoiterivin numero vastaa Aromin tietokannan ateriatunnusta.

Esimerkki: <http://redlabs.laurea.fi/ATPSite/?value=5180>

WWW-osoitteessa oleva numero 5180 siis määrää täysin käyttäjälle näytettävän sivun sisällön. Mobiili-ATP sijaitsee Laurea Leppävaaran RedLabsin palvelimella ja on käytettävissä kaikkialta ilman rajoituksia. Mobiili-ATP:n kommentoitu lähdekoodi on liitteessä 1.

Ravintoarvot (100g):			
Hiilihydraatit (g)	Rasvat (g)	Proteiinit (g)	Kalorit (kcal)
3	6	7	94

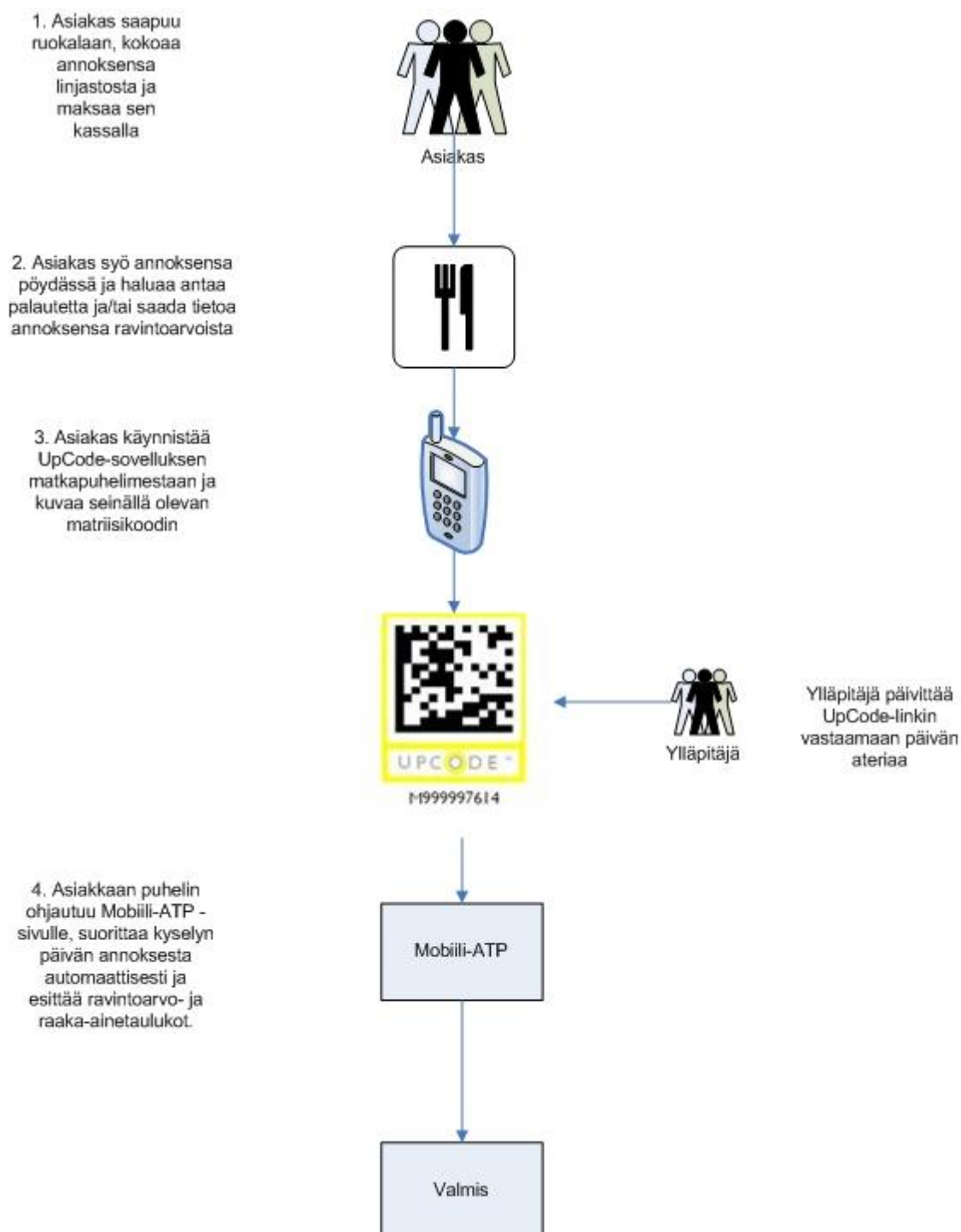
Raaka-aineet:	
Ilmi	
Rypsiöljy	
Pomodoro Rosso, Apetit	
Vesi	
Paistinkastiketiiviste, Kulta Sula	
Pizzakastike, maustettu säilyke	
Kasvisliemijauhe	
Sokeri, hieno	
Suola, ruoka	
Pippuri, seos	
Oregano, kuivattu	
Basilika	
Jauheliha, paistettu	

Kuva 19: Mobiili-ATP

Kuvassa 19 on esimerkki mobiili-ATP:n sisällöstä. Ylempässä taulukossa nähdään annoksen ravintoarvot, jotka koostuvat hiilihydraateista, rasvasta, proteiineista ja kaloreista. Alempi taulukko erittelee annoksessa käytetyt raaka-aineet. Sivun suunnittelussa on pyritty yksinkertaisuuteen ja selkeyteen. Tiedonsiirron halutaan olevan mahdollisimman nopeaa koska käyttö on tarkoitettu tapahtuvan mobiililaitteella.

6.2 Mobiili-ATP:n käyttö

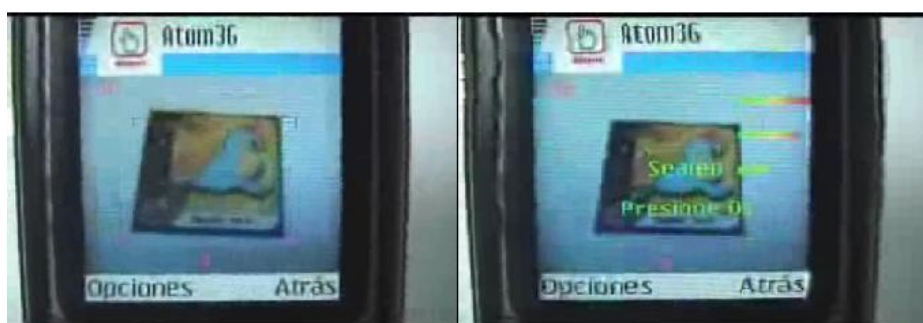
Kuvassa 20 esitetty käyttötapaus kuvaa Mobiili-ATP:n suunniteltua käyttöä. Asiakas saapuu ruokalaan. Ruokalan seinällä, pöydässä tai ruokalistassa on tarjolla olevia ruokia vastaavat koodit. Asiakas käynnistää puhelimestaan UpCode-sovelluksen ja kuvaa ateriaansa vastaavan koodin. Tämä ohjaa puhelimen internetselaimen ylläpitäjän määrittämälle sivulle, ja asiakas saa välittömästi puhelimensa ruudulle ravintoarvotiedot valitsemastaan ateriatesta.



Kuva 20: Mobiili-ATP:n käyttötapauskaavio

7 Matriisikoodien tulevaisuudennäkymiä (ATOM TAG)

Barcelonalainen DAEM Interactive on kehittänyt uudenlaisen kuvantunnistusteknologian nimeltä ATOM TAG. Tähän tarvitaan 3G-yhteydellä varustettu kamerapuhelin, ja ATOM TAG -ohjelmisto. ATOM TAG eroaa muista matriisikoodista siten, että koodit eivät ole varsinaisia koodoja, sillä ne voivat olla esimerkiksi firmojen logoja. Uutta tässä on myös se, että koodin luettuaan informaatio ilmestyy suoraan mobiililaitteen ruudulle, kamerakuvan päälle. Kyseessä on niin sanottu "Augmented Reality" eli laajennettu todellisuus, jossa käyttäjä näkee esimerkiksi kamerapuhelimensa läpi tekstiä, jota paljaalla silmällä katsoessa ei näkyisi. (Atom Tag 2009.) Kuvassa 21 on esimerkki Atom TAG -teknologian käytöstä.



Kuva 21: Vasemmalla logo on ennen tunnistusta ja oikealla tunnistuksen jälkeen.

8 Tulokset ja johtopäätökset

Työssä tutustuttiin matriisikodeihin ja niiden käyttötarkoituksiin. Tekstiosuuden tavoitteena oli laajentaa lukijan tietoja aiheesta. Toiminnallinen osuus eli Mobiili-ATP hyödyntää vuoden 2006 Mikko Lassilan "Ateriatietopalvelun kehittäminen ja käyttöönotto BarLaureassa" -opinnäytetyötä ja sen puitteissa onnistuu tehtävässään mielestämme varsin hyvin. Annoksen nimeä emme saaneet näkyviin ravintotietojen yhteyteen, koska Mobiili-ATP:n käyttämä WebService-rajapinta ei tarjoa mahdollisuutta siihen. Tämä voidaan kuitenkin ratkaista niin, että merkitään päivän ruokalistaan jokaisen annoksen viereen sitä vastaava koodi. Tällöin asiakkaalle pitäisi olla selvää, minkä annoksen ravintoarvoja hän hakee. Myös raaka-ainelista kertoo asiakkaalle, katsooko hän oikean annoksen ravintoarvoja.

Mobiili-ATP:n toteutus oli haastavaa, koska kummallakaan työn tekijöistä ei ole mainittavia koodaustaitoja. Laurean kirjastosta löytynyt ohjelmointikirjallisuus, MSDN (Microsoft Developer Network) ja lukuisat internetistä löytyneet esimerkit avustivat ohjelman luonnissa suuresti.

Matriisikoodit ovat uusi teknologia, josta on vaikea löytää painettua tietoa. Tästä syystä lähes kaikki opinnäytetyön lähteet ovat nettiviitteitä. Arvioimme pohjatietojemme ja lähteiden luotettavuuden perusteella kuitenkin lähteet luotettaviksi, ja siksi päätimme käyttää niitä.

Uskomme, että matriisikoodit tulevat yleistymään entisestään, tosin ei välttämättä nykyisessä mustavalkoisessa muodossaan. Tulevaisuudessa tulemme ehkä näkemään uusia koodityyppejä ja niiden sovelluksia. Innovatiiviset kehittäjät kuten ATOM Tag -teknologian DAEM Interactive osoittavat, että tunnistusteknologian ei tarvitse rajoittua ainoastaan matriisikodeihin.

Kenties näemme jonakin päivänä laajennettu todellisuus -silmlaseja, jotka antavat käyttäjälleen lisätietoa lähes kaikesta mitä käyttäjä näkee.

Sanasto

ASCII-merkistö

ASCII (lyhenne sanoista American Standard Code for Information Interchange) on tietokoneiden merkistö, joka sisältää englannin kielen kirjaimet, numerot, joukon välimerkkejä ja joitakin ohjauskoodeja.

C# / .NET -ympäristö

C# (C Sharp) on ohjelmointikieli jota voidaan käyttää .NETin kanssa. .NET on Windowsin osa joka sisältää paljon valmiita ratkaisuja ohjelmointiin kuten järjestelmien välinen kommunikointi alustasta riippumatta.

EAN -Formaatti

EAN-koodi eli eurooppalainen artikkelinumero (European Article Number) tarkoittaa tapaa koodata tuotteet siten, etteivät ne mene sekaisin keskenään. Koodi muodostuu kahdesta osasta, tuotteen yksilöivästä numerosta ja numeroa vastaavasta koneellisesti luettavasta symbolista, viivakoodista.

ECC200-virheenkorjaus

ECC viittaa sanoihin "error correction code". ECC200 on matriisikoodien koodaustapa, joka sallii tiedon säilymisen ehjänä vaikka 30% matriisista on vahingoittunut.

ISO/IEC 16022 -standardi

ISO/IEC 16022:2006 määrittää vaatimukset Data Matrix -symbolologialle. Se standardisoi merkistön, koodikielen, symbolien formaatit, koon ja tulostamistavat, sekä virheenkorjausmenetelmät, purkumenetelmät, ja käyttäjän päätettävissä olevat parametrit.

Kaksiulotteinen viivakoodi

Matriisikoodi on kaksiulotteinen viivakoodi, eli symboli joka sisältää mustia ja valkoisia "soluja", eli moduuleita, yleensä järjestettynä joko neliö- tai suorakaiteen muotoiseksi. Symboliin

	koodattava tieto voi olla tekstiä tai muuta tietoa (nettilinkit, tuotetiedot jne.)
Marginaali (quiet zone border)	Marginaalilla tarkoitetaan tässä yhteydessä viivoja jotka jakavat matriisisymbolin esimerkiksi neljään eri osaan.
Open source -koodi	Matriisikoodi, joka on vapaasti tuotettavissa ja muokattavissa.
Reed-Solomon -virheenkorjaus	Reed-Solomon -virheenkorjausmetodi on virheenkorjauskoodi, joka toimii monistamalla koodin sisältämästä datasta tehtyä polynomia, eli matemaattista lauseketta.
Solu	Matriisikoodin sisältämä musta tai valkoinen neliö.
UPC	UPC-koodi (Universal Product Code) on viivakoodityyppi, joka on laajalti käytössä Kanadassa ja Yhdysvalloissa tuotteiden seurannassa.
Upcode	UpCode on suomalaisen UpCode -yrityksen kehittämä matriisikoodi.
WebService	Ohjelmistojärjestelmä joka mahdollistaa keskenään yhteensopivien tietokoneiden välisen vuorovaikutuksen tietoverkon yli.

Lähteet

Atom Tag. 2009. Viitattu 20.11.2009.

<http://www.gotomobile.com/archives/next-generation-2d-code-atom-tag>

Aztec. 2009. Wikipedia. Viitattu 20.11.2009.

http://en.wikipedia.org/wiki/Aztec_code

Datamatriisit. 2009. Wikipedia. Viitattu 20.11.2009.

http://en.wikipedia.org/wiki/Data_matrix_%28computer%29

Denim Code. 2009. Viitattu 20.11.2009.

<http://www.denimcode.fr/fr/new/denimcode-tag.php>

Facebook. 2009. Viitattu 20.11.2009.

<http://www.facebook.com/apps/application.php?id=21352510322>

Lahti, J. 2008. UpCode-älyruutusovellus etenee maailmalla. IT-viikko. Viitattu 20.11.2009.

<http://www.itviikko.fi/teknologia/2008/03/26/upcode-alyruutusovellus-etenee-maailmalla/20088568/7>

Järvinen, P. & Järvinen, A. 2000. Tutkimustyön metodeista. 2. painos. Tampere: Yliopistopaino.

Lassila, M. 2006. Ateriatietopalvelun kehittäminen ja käyttöönotto BarLaureassa. Opinnäytetyö. Laurea-ammattikorkeakoulu.

Microsoft Tag. 2009. Wikipedia. Viitattu 20.11.2009.

http://en.wikipedia.org/wiki/Microsoft_Tag#Microsoft_Tag

Object Hyperlinking. 2009. Wikipedia. Viitattu 20.11.2009.

http://en.wikipedia.org/wiki/Object_hyperlinking

PDF417. 2009. Wikipedia. Viitattu 20.11.2009.

<http://en.wikipedia.org/wiki/PDF417>

Ojanperä, V. 2008. HKL:n kaikille pysäkeille Upcode-älyruudut. Prosessori. Viitattu 20.11.2009.

<http://www.prosessori.fi/uutiset/uutinen2.asp?id=52669>

QR-codes explained. 2009. Viitattu 20.11.2009.

<http://www.qrme.co.uk/qr-codes-explained.html>

QR-koodi. 2009. Wikipedia. Viitattu 20.11.2009.

http://en.wikipedia.org/wiki/QR_Code

Semacode. 2009. Wikipedia. Viitattu 20.11.2009.

<http://en.wikipedia.org/wiki/Semacode>

Shotcode. 2009. Wikipedia. Viitattu 20.11.2009.

<http://en.wikipedia.org/wiki/ShotCode>

UpCode. 2009. Upc. Viitattu 20.11.2009.

<http://www.upc.fi/en/upcode/company/>

Kuvat

Kuva 1: Viivakoodi ja sen lukeminen	5
Kuva 2: UpCode-matriisikoodi	6
Kuva 3: Kaukaa kuvattava matriisikoodi	7
Kuva 4: Objektien hyperlinkittäminen	10
Kuva 5: Semacode-koodi	12
Kuva 6: 4-osainen Datamatrix-koodi	12
Kuva 7: Shotcode-koodi	13
Kuva 8: PDF417-koodi	13
Kuva 9: Aztec-koodi	14
Kuva 10: QR-koodi	14
Kuva 11: Microsoft Tag -koodeja	15
Kuva 12: Microsoft Tag -koodi verrattuna QR- ja Datamatrix-koodeihin	16
Kuva 13: UpCode-koodi	16
Kuva 14: Add to Friends -T-paita QR-koodilla	17
Kuva 15: Housut, joissa matriisikoodi	18
Kuva 16: Abukuma Chicken Farm -kanala	19
Kuva 17: Abukuma Chicken Farm -kananmunia	19
Kuva 18: Kyselyiden suoritus Webservicen kautta Aromista	20
Kuva 19: Mobiili-ATP	21
Kuva 20: Mobiili-ATP:n käyttötapauskaavio	22
Kuva 21: Vasemmalla logo on ennen tunnistusta ja oikealla tunnistuksen jälkeen.	23

Liitteet

Liite 1: Mobiili-ATP:n lähdekoodi

30

Mobiili-ATP:n lähdekoodi

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;

namespace ATPSite
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                // Katsotaan että sivun parametrejä on vähintään yksi
                if (Page.Request.Params.Count > 0)
                {
                    // Luetaan URL:stä parametria value, mikäli parametria ei löydy, ohjelma
                    // menee exceptioniin
                    string value = Page.Request.Params["value"].ToString();

                    // Luodaan WebService-olio, ATPWebService-viittauksesta
                    // viittauksen luonti (Add Web Reference), luo automaattisesti kaikki
                    // tarvittavat metodit ja oliot WebServiceen käyttöä varten.
                    WS_ATP.ATPWebService service = new ATPSite.WS_ATP.ATPWebService();

                    // Asetetaan DataGrid-komponentille teksti sen varalta, että data jää tyhjäksi
                    dgvRavintoArvot.EmptyDataText = "Haualla ei löytynyt yhtään dataa tietokannasta";

                    // Haetaan ravintoarvot ID:llä, mikä on saatu value parametrinä
                    WS_ATP.Ravintoarvot values = service.HaeRavintoarvot(value);

                    // Luodaan DataTable johon tieto säilötään.
                    DataTable table = new DataTable();
```

```

// Laitetaan halutut sarakkeet datatablelle. Nämä sarakkeet näytetään
// DataGridissä.
table.Columns.Add("Hiilihydraatit (g)");
table.Columns.Add("Rasvat (g)");
table.Columns.Add("Proteiinit (g)");
table.Columns.Add("Kalorit (g)");

// Loopataan sarakkeet läpi ja asetetaan tietotyyppi int
foreach (DataColumn col in table.Columns)
{
    col.DataType = typeof(int);
}

// Syötetään ravintoarvot DataGridiin. Objektitaulumuoto linkittyy
// automaattisesti oikeisiin kenttiin
table.Rows.Add(new object[] { values.hiilihydraatit, values.rasvat,
values.proteiinit, values.kalorit });

// Linkitetään data
dgvRavintoArvot.DataSource = table;
dgvRavintoArvot.DataBind();

// Haetaan raaka-aineet ID:llä
WS_ATP.RaakaAineet[] raakaaineet = service.HaeRaakaAineet(value);

// Luodaan toinen DataTable
DataTable raakaaineTable = new DataTable();

// Lisätään vain Nimisarake. Muita ei haluta nähdä
raakaaineTable.Columns.Add("Nimi");

// Käydään läpi kaikki raaka-aineet mitä on löytynyt.
foreach (WS_ATP.RaakaAineet aine in raakaaineet)
{
    // Lisätään raaka-aineen nimi DataTableeen
    raakaaineTable.Rows.Add(new object[] { aine.nimi });
}

```



```
        // Linkitetään DataGrid komponenttiin
        dgvRaakaAineet.DataSource = raakaaineTable;
        dgvRaakaAineet.DataBind();
    }
}
catch
{
    // Ei virheen käsittelyä
}
}
}
}
```